IN THE CLAIMS

1. (Amended) A method for context switching a processor that executes procedures having differing word sizes, comprising [the steps of]:

testing a most significant bit of a stack pointer register in the processor that indicates whether a set of data values for a procedure that are stored in a set of registers in the processor each have a first word size or a second word size wherein the first word size is less than the second word size;

transferring the data values from a least significant portion of each register to a first stack save area in memory and transferring a least significant portion of a stack pointer value from the stack pointer register to the first stack save area in memory if the most significant bit of the stack pointer register indicates the first word size;

setting a width indication bit in the first stack save area in memory, and transferring the data values from the registers to a second stack save area in memory and transferring the stack pointer value from the stack pointer register to the second stack save area if the most significant bit of the stack pointer register indicates the second word size such that the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

2. (Amended) The method of claim 1, further comprising [the steps of]:

testing the width indication bit in the first stack save area in memory;

reading the data values and the least significant portion of the stack pointer value from the first stack save area, and storing the data values into the least significant portions of the registers and storing the least significant portion of the stack pointer value into the stack pointer register, and clearing the most significant bit of the stack pointer register to indicate that the data values for the procedure have the first word size if the width indication bit in the first stack save area in memory does not indicate that the data values for the procedure have the second word size;

reading the data values and the stack pointer value from the second stack save area, and storing the data values into the registers and storing the stack pointer value in the stack pointer register if the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

3. (Original) The method of claim 2, wherein the width indication bit in the first stack save area comprises a least significant bit in a location of the first stack save area allocated to the stack pointer value for the procedure.

4. (Original) The method of claim 2, wherein the first stack save area is specified by the stack pointer value in the stack pointer register.

5. (Original) The method of claim 4, wherein the second stack save area is specified by the stack pointer value in the stack pointer register plus an offset value that corresponds to an area in memory required for the first stack save area.

6. (Original) The method of claim 5, wherein the first word size comprises 32 bits and the second word size comprises 64 bits.

7. (Original) The method of claim 6, wherein the registers in the processor and the stack pointer register in the processor comprise 16 registers each comprising 64 bits.

8. (Original) The method of claim 7, wherein the offset value and the area in memory for the first stack save area each comprise 16 multiplied by 4 bytes per register.

9. (Original) A processor that executes procedures having differing word sizes, comprising:

means for testing a most significant bit of a stack pointer register in the processor that indicates whether a set of data values for a procedure that are stored in a set of registers in the processor each have a first word size or a second word size wherein the first word size is less than the second word size;

means for transferring the data values from a least significant portion of each register to a first stack save area in memory and transferring a least significant portion of a stack pointer value from the stack pointer register to the first stack save area in memory if the most significant bit of the stack pointer register indicates the first word size;

means for setting a width indication bit in the first stack save area in memory, and transferring the data values from the registers to a second stack save area in memory and transferring the stack pointer value from the stack

pointer register to the second stack save area if the most significant bit of the stack pointer register indicates the second word size such that the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

10. (Original) The processor of claim 9, further comprising:

means for testing the width indication bit in the first stack save area in memory;

means for reading the data values and the least significant portion of the stack pointer value from the first stack save area, and storing the data values into the least significant portions of the registers and storing the least significant portion of the stack pointer value into the stack pointer register, and clearing the most significant bit of the stack pointer register to indicate that the data values for the procedure have the first word size if the width indication bit in the first stack save area in memory does not indicate that the data values for the procedure have the second word size;

means for reading the data values and the stack pointer value from the second stack save area, and storing the data values into the registers and storing the stack pointer value in the stack pointer register if the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

11. (Original) The processor of claim 10, wherein the width indication bit in the first stack save area comprises a least significant bit in a location of the first stack save area allocated to the stack pointer value for the procedure.

12. (Original) The processor of claim 10, wherein the first stack save area is specified by the stack pointer value in the stack pointer register.

13. (Original) The processor of claim 12, wherein the second stack save area is specified by the stack pointer value in the stack pointer register plus an offset value that corresponds to an area in memory required for the first stack save area.

14. (Original) The processor of claim 13, wherein the first word size comprises 32 bits and the second word size comprises 64 bits.

15. (Original) The processor of claim 14, wherein the registers in the processor and the stack pointer register in the processor comprise 16 registers each comprising 64 bits.

16. (Original) The processor of claim 15, wherein the offset value and the area in memory for the first stack save area each comprise 16 multiplied by 4 bytes per register.

17. (Amended) A method for context switching a processor that executes procedures having differing word sizes, comprising [the steps of]:

testing a least significant bit of a stack pointer register in the processor that indicates whether a set of data values for a procedure that are stored in a set of registers in the processor each have a first word size or a second word size wherein the first word size is less than the second word size;

transferring the data values from a least significant portion of each register to a first stack save area in memory and transferring a least significant portion of a stack pointer value from the stack pointer register to the first stack save area in memory if the least significant bit of the stack pointer register indicates the first word size;

setting a width indication bit in the first stack save area in memory, and transferring the data values from the registers to a second stack save area in memory and transferring the stack pointer value from the stack pointer register to the second stack save area if the least significant bit of the stack pointer register indicates the second word size such that the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

18. (Amended) The method of claim 17, further comprising [the steps of]:

testing the width indication bit in the first stack save area in memory;

reading the data values and the least significant portion of the stack pointer value from the first stack save area, and storing the data values into the least significant portions of the registers and storing the least significant portion of the stack pointer value into the stack pointer register, and clearing the least significant bit of the stack pointer register to indicate that the data values for the procedure have the first word size if the width indication bit in the first stack save area in memory does not indicate that the data values for the procedure have the second word size;

reading the data values and the stack pointer value from the second stack save area, and storing the data values into the registers and storing the stack pointer value in the stack pointer register if the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

19. (Previously Presented) The method of claim 18, wherein the width indication bit in the first stack save area comprises a least significant bit in a location of the first stack save area allocated to the stack pointer value for the procedure.

20. (Previously Presented) The method of claim 18, wherein the first stack save area is specified by the stack pointer value in the stack pointer register.

21. (Previously Presented) The method of claim 20, wherein the second stack save area is specified by the stack pointer value in the stack pointer register plus an offset value that corresponds to an area in memory required for the first stack save area.

22. (Previously Presented) The method of claim 21, wherein the first word size comprises 32 bits and the second word size comprises 64 bits.

23. (Previously Presented) The method of claim 22, wherein the registers in the processor and the stack pointer register in the processor comprise 16 registers each comprising 64 bits.

24. (Previously Presented)The method of claim 23, wherein the offset value and the area in memory for the first stack save area each comprise 16 multiplied by 4 bytes per register.

25. (Previously Presented) A processor that executes procedures having differing word sizes, comprising:

means for testing a least significant bit of a stack pointer register in the processor that indicates whether a set of data values for a procedure that are stored in a set of registers in the processor each have a first word size or a second word size wherein the first word size is less than the second word size;

means for transferring the data values from a least significant portion of each register to a first stack save area in memory and transferring a least significant portion of a stack pointer value from the stack pointer register to the first stack save area in memory if the least significant bit of the stack pointer register indicates the first word size;

means for setting a width indication bit in the first stack save area in memory, and transferring the data values from the registers to a second stack save area in memory and transferring the stack pointer value from the stack pointer register to the second stack save area if the least significant bit of the stack pointer register indicates the second word size such that the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

26. (Previously Presented) The processor of claim 25, further comprising:

means for testing the width indication bit in the first stack save area in memory;

means for reading the data values and the least significant portion of the stack pointer value from the first stack save area, and storing the data values into the least significant portions of the registers and storing the least significant portion of the stack pointer value into the stack pointer register, and clearing the least significant bit of the stack pointer register to indicate that the data values for the procedure have the first word size if the width indication bit in the first stack save area in memory does not indicate that the data values for the procedure have the second word size;

means for reading the data values and the stack pointer value from the second stack save area, and storing the data values into the registers and storing the stack pointer value in the stack pointer register if the width indication bit in the first stack save area in memory indicates that the data values for the procedure have the second word size.

27. (Previously Presented) The processor of claim 26, wherein the width indication bit in the first stack save area comprises a least significant bit in a location of the first stack save area allocated to the stack pointer value for the procedure.

28. (Previously Presented) The processor of claim 26, wherein the first stack save area is specified by the stack pointer value in the stack pointer register.

29. (Previously Presented) The processor of claim 28, wherein the second stack save area is specified by the stack pointer value in the stack pointer register plus an offset value that corresponds to an area in memory required for the first stack save area.

30. (Previously Presented) The processor of claim 29, wherein the first word size comprises 32 bits and the second word size comprises 64 bits.

31. (Previously Presented) The processor of claim 30, wherein the registers in the processor and the stack pointer register in the processor comprise 16 registers each comprising 64 bits.

32. (Previously Presented) The processor of claim 31, wherein the offset value and the area in memory for the first stack save area each comprise 16 multiplied by 4 bytes per register.

33. (Twice Amended) A method for context switching a processor that executes procedures having differing word sizes, comprising [the steps of]:

testing a least significant bit of a stack pointer register in the processor that indicates whether a set of data values for a procedure that are stored in a set of registers in the processor each have a first word size or a second word size wherein the first word size is less than the second word size;

transferring the data values from a least significant portion of each register to a first stack save area in memory if the least significant bit of the stack pointer register indicates the first word size;

setting a width indication bit in the first stack save area in memory, and transferring the data values from the registers to a second stack save area in memory if the least significant bit of the stack pointer register indicates the second word size.

34. (Previously Presented) The method of claim 33, wherein the first stack save area is specified by the stack pointer value in the stack pointer register.

35. (Previously Presented) The method of claim 34, wherein the second stack save area is specified by the stack pointer value in the stack pointer register plus an offset value that corresponds to an area in memory required for the first stack save area.

36. (Previously Presented) The method of claim 33, wherein the first word size comprises 32 bits and the second word size comprises 64 bits.

37. (Previously Presented) The method of claim 33, wherein the registers in the processor and the stack pointer register in the processor comprise 16 registers each comprising 64 bits.

38. (Previously Presented) The method of claim 35, wherein the offset value and the area in memory for the first stack save area each comprise 16 multiplied by 4 bytes per register.